My Perspectives on Operating SDN Networks

Shawn McKee/University of Michigan

Roadmap to Operating SDN-based Networks

Berkeley, CA

July 14th 2015

Introduction

✴ For those that don't know me a quick introduction:

- ❑ *Shawn McKee, University of Michigan Physics*
- ❑ *Director of the ATLAS Great Lakes Tier-2 (AGLT2)*
- ❑ *USATLAS Network Manager*
- ❑ *Open Science Grid Network Area Coordinator*
- ❑ *WLCG Network and Transfer Metrics Working Group Co-Chair*
- ❑ *Co-PI on current and past network-related NSF projects: Ultralight, PLaNetS, DYNES, ANSE, PuNDIT*

✴ *My longstanding network interest is motivated by supporting LHC distributed, data-intensive science needs.*

- ❑ *I would like to see our networks becomes a managed component of our infrastructure, similar to what we have for compute and storage.*

✴ *The comments in my presentation incorporate input/ feedback from Kaushik De/UTA and Ilija Vukotic/UC, both colleagues from ATLAS*

Context for My Perspective

✳ Integrating the network into a managed infrastructure is still "hard" to do because:

- ❏ Network hardware vendors haven't (yet) provided production quality SDN components

- ❏ What services and capabilities exist along a given path?

- ❏ I setup my SDN network (path or topology) and data isn't flowing. What do I do?

- ❏ Am I getting what I asked for? How do I monitor it?

- ❏ Network-view of data-flows is in a different language/context compared to application level view. How do we bridge this gap to allow better integration and use of SDN?

Network Offloading

✳ Networks historically have been black-boxes where applications/users stuff bits in one end and hope they come out when and where they are needed.

✳ As SDN as evolved we have the promise of creating a managed integrated network controllable to optimize the overall system for our needs.

❑ In practice this has been problematic, in part because of the level of knowledge required by the SDN-users about networking.

✳ I would like to see a cohesively designed SDN that off-loads and organizes details transparently for end-users

❑ For example, can an SDN framework automate tracking and managing specific flows associated with specific tasks in the context of what the end-application understands?

❑ Could the application then request priority for certain workflows it is managing and have the network respond accordingly?

SDN Wishlist

✳ Any SDN framework intended to be used for data-intensive distributed science must have **monitoring** and **debugging** built-in as first-class components from the start

- ❑ Any SDN API should automatically provide the means to monitor and query components created, organized or assembled by SDN, preferably as part of a system level design.
- ❑ A coherent debugging framework should exist for the SDN components and entities.

✳ Example: If I create a point-to-point SDN circuit, the object representing that circuit should have a method to request monitoring which may include access to counters from devices along the path and/or active and/or passive monitoring of the traffic handling capabilities and characteristics of the path. Likewise, when the data plane is not passing traffic I want access to debugging details along the path which will allow localization of the data-plane failure location(s).

✳ **Discovery of services and capabilities must be in place. I need to know what my options are on any given source-destination path.**

Challenges for Using SDN

✴ I have seen lots of challenges trying to deploy and use SDN capabilities.

✴ **Getting capability end-to-end is always a challenge**. Typically end up "tunneling" through (via VLAN) non-SDN paths. Impact on end-to-end behavior is hard to quantify

  ❑ Getting all the way to the "end" is hard. We want to have SDN from storage-to-storage, computer-to-computer, application-to-application

✴ OpenFlow Example: Vendors deliver OpenFlow 1.x* on their hardware and we try to use it.

  ❑ Read the fine print.   Note the exceptions, caveats and gotcha's

  ❑ Google for other's experiences to find out how things actually work (or don't)

  ❑ For the above reasons, we need to run a segregated SDN network from our typical production network.   This prevents making real progress in getting SDN into "Production".

Opportunities for SDN

✴ Network vendors are starting to get things right.

  ❑ New hardware supporting SDN needs on silicon.

  ❑ More extensive commercial use of SDN leading to better tested and working implementations

  ❑ Expectation is that within the next 2-3 years we should see significant deployment of "production quality" SDN along many of our R&E paths

✴ Virtualization efforts extending beyond computing and storage and including the network.

  ❑ Projects like OpenvSwitch (see http://openvswitch.org) can help us get SDN to end-hosts and applications; integrate WAN and ends.

✴ Things are <u>not yet</u> broadly deployed. We still have time to influence what future SDN production networks will look like

Summary

✱ To use SDN as part of our distributed science infrastructure we need:

- ❑ Discovery of services, topology and capabilities along all our end-to-end paths
- ❑ Monitoring of the network at many levels to inform both users and services about how things are working.   (Did I get what I asked for?)
- ❑ Integrated debugging, designed in at the "system" level. When something isn't working we need the tools to locate the problem ASAP.
- ❑ Pervasive deployment of SDN-capable **production** hardware
- ❑ Application-level awareness and integration of SDN

✱ I believe the application and infrastructure middleware developers would love to have production-quality SDN, allowing incorporation of the network into their distributed infrastructure; it just needs to be more straightforward to do.